AZTEKAS: a hydrodynamic GPL code

Version1.0

Generated by Doxygen 1.8.16

1 AZTEKAS: a hydrodynamic GPL code 1

2 File Index 3

2.1 File List . . . 3

3 File Documentation 5

3.1 alloc.c File Reference . . . 5

3.1.1 * . . . 5

3.1.2 Detailed Description . . . 5

3.2 array.c File Reference . . . 6

3.2.1 * . . . 6

3.2.2 Detailed Description . . . 6

3.3 auxfunc.c File Reference . . . 6

3.3.1 * . . . 6

3.3.2 Detailed Description . . . 7

3.4 bound_cond.c File Reference . . . 7

3.4.1 * . . . 7

3.4.2 Detailed Description . . . 7

3.5 flux.c File Reference . . . 7

3.5.1 * . . . 7

3.5.2 Detailed Description . . . 8

3.6 input.c File Reference . . . 8

3.6.1 * . . . 8

3.6.2 * . . . 8

3.6.3 Detailed Description . . . 8

3.7 integration.c File Reference . . . 8

3.7.1 * . . . 8

3.7.2 Detailed Description . . . 9

3.8 main.c File Reference . . . 9

3.8.1 * . . . 9

3.8.2 Detailed Description . . . 9

3.9 output.c File Reference . . . 9

3.9.1 * . . . 9

3.9.2 Detailed Description . . . 10

3.10 restart.c File Reference . . . 10

3.10.1 * . . . 10

3.10.2 Detailed Description . . . 10

3.11 timestep.c File Reference . . . 10

3.11.1 * . . . 10

3.11.2 * . . . 11

3.11.3 Detailed Description . . . 11

Index 13

Chapter 1

AZTEKAS: a hydrodynamic GPL code

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

alloc.c
Essential allocation functions for *aztekas* . . . 5
array.c
Functions to simplify the index vector access . . . 6
auxfunc.c
Helpful functions for *aztekas* . . . 6
bound_cond.c
Standard boundary conditions . . . 7
flux.c
Numerical flux computing and implementation . . . 7
input.c
Important input parameters for *aztekas* . . . 8
integration.c
Main function for the time integration in the conservative variables $\mathbf{Q}$ . . . 8
main.c
Main file of aztekas . . . 9
output.c
Output functions: ASCII and Binary . . . 9
restart.c
Functions to restart from a given file . . . 10
timestep.c
Time-step calculation . . . 10

Chapter 3

File Documentation

3.1 alloc.c File Reference

Essential allocation functions for *aztekas*.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.1.1 *

Functions

- void **allocateArray** ()
- void **new_SIZE** ()

3.1.2 Detailed Description

Essential allocation functions for *aztekas*.

Author

Alejandro Aguayo-Ortiz

3.2 array.c File Reference

Functions to simplify the index vector access.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.2.1 *

Functions

- int **c1** (int n, int i)
- int **c2** (int n, int i, int j)
- int **c3** (int n, int i, int j, int k)

3.2.2 Detailed Description

Functions to simplify the index vector access.

Author

Alejandro Aguayo-Ortiz

In this file we include three functions for passing the standard C notation for a vector:
`U[i*N_j*N_k + j*N_k + k]`

to a much simpler notation
`U[(i,j,k)]`

3.3 auxfunc.c File Reference

Helpful functions for *aztekas*.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.3.1 *

Functions

- int **MxV** (double *M, double *V, double *L)
- void **roundgen** (double *num)

3.3.2 Detailed Description

Helpful functions for *aztekas*.

Author

Alejandro Aguayo-Ortiz

3.4 bound_cond.c File Reference

Standard boundary conditions.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.4.1 *

Functions

- void **OUTFLOW** (double *B)
- void **REFLECTIVE** (double *B)
- void **PERIODIC** (double *B)

3.4.2 Detailed Description

Standard boundary conditions.

Author

Alejandro Aguayo-Ortiz

3.5 flux.c File Reference

Numerical flux computing and implementation.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "param.h"
#include "main.h"
```

3.5.1 *

Functions

- int **FLUX1D** (vec_ *v, lim_ *l, int *l)
- int **FLUX2D** (vec_ *v, lim_ *l, int *l)
- int **FLUX3D** (vec_ *v, lim_ *l, int *l)
- int **HLL** (double *F, flx_ *f, int x)
- int **HLLC** (double *F, flx_ *f, int x)

3.5.2 Detailed Description

Numerical flux computing and implementation.

Author

Alejandro Aguayo-Ortiz

3.6 input.c File Reference

Important input parameters for *aztekas*.

```
#include <stdio.h>
#include <omp.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
```

3.6.1 *

Functions

- int **read_parameters_file** (char const ∗paramfile_name)

3.6.2 *

Variables

- FILE ∗ **paramfile**

3.6.3 Detailed Description

Important input parameters for *aztekas*.

Author

Emilio Tejeda

3.7 integration.c File Reference

Main function for the time integration in the conservative variables $\mathbf{Q}$.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "vector.h"
```

3.7.1 *

Functions

- int **INTEGRATION** ()

3.7.2 Detailed Description

Main function for the time integration in the conservative variables $\mathbf{Q}$.

Author

Alejandro Aguayo-Ortiz

3.8 main.c File Reference

Main file of aztekas.

```
#include <stdio.h>
#include <omp.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.8.1 *

Functions

- int **main** (int argc, char *argv[ ])

3.8.2 Detailed Description

Main file of aztekas.

Author

Alejandro Aguayo-Ortiz.

3.9 output.c File Reference

Output functions: ASCII and Binary.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "main.h"
```

3.9.1 *

Functions

- int **PrintValues** (double *tprint, double *dtprint, int *itprint)
- int **Output1** (int *itprint)
- int **Output1_bin** (int *itprint)
- int **Output2** (int *itprint)
- int **Output2_bin** (int *itprint)
- int **Output3** (int *itprint)
- int **Output3_bin** (int *itprint)

3.9.2 Detailed Description

Output functions: ASCII and Binary.

Authors

Alejandro Aguayo-Oritz and Emilio Tejeda

3.10 restart.c File Reference

Functions to restart from a given file.

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include "main.h"
#include "param.h"
```

3.10.1 *

Functions

- void **RESTART** ()
- void **RESTART_BIN** ()

3.10.2 Detailed Description

Functions to restart from a given file.

Author

Emilio Tejeda

3.11 timestep.c File Reference

Time-step calculation.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "./Headers/main.h"
```

3.11.1 *

Macros

- #define **min**(a, b) (((a)<(b))?(a):(b))
- #define **max**(a, b) (((a)>(b))?(a):(b))

3.11.2 *

Functions

- double **TIMESTEP** ()

3.11.3 Detailed Description

Time-step calculation.

Author

Alejandro Aguayo-Ortiz

Index

alloc.c, 5
array.c, 6
auxfunc.c, 6

bound_cond.c, 7

flux.c, 7

input.c, 8
integration.c, 8

main.c, 9

output.c, 9

restart.c, 10

timestep.c, 10